

Programa del curso IC-6831

Aseguramiento de la calidad del software

Escuela de Computación
Ingeniería en Computación, Plan 411
Modalidad Verano – 6 semanas

I parte: Aspectos relativos al plan de estudios

I Datos generales

Nombre del curso:	Aseguramiento de la calidad del software.
Código:	IC-6831
Tipo de curso:	Teórico - Práctico
Nº de créditos:	3
Nº horas de clase por semana:	3
Nº horas extra-clase por semana:	6
Ubicación en el plan de estudios:	Curso del sexto semestre de la carrera de Ingeniería en Computación
Requisitos:	IC-6821 Diseño de software IC-4810 Administración de proyectos.
Correquisitos:	Ninguno
El curso es requisito de:	IC-7841 Proyecto de Ingeniería del software
Asistencia:	Obligatoria
Suficiencia:	Sí.
Posibilidad de reconocimiento:	Sí.
Vigencia del programa:	Segundo semestre del 2020 en adelante.

2 Descripción general

La evaluación de la calidad del software vía verificación y validación puede ahorrar a una organización esfuerzo, tiempo y dinero. Las mejoras en la calidad del software implican usualmente mejoras en la productividad también. Este curso presenta una perspectiva pragmática del control de la calidad del software enmarcado en procesos de ingeniería del software, con énfasis en las técnicas de pruebas y revisiones estáticas, así como los procesos y las actividades del aseguramiento de la calidad.

3 Objetivos

Objetivo general

Experimentar los procesos de calidad del software mediante la realización de actividades de planeación, validación y verificación, para la producción de artefactos y sistemas que satisfagan los requerimientos en forma oportuna y económica.

Objetivos específicos

- Describir las características de la calidad del software, mediante la revisión de modelos pertinentes, para la comprensión de su importancia - así como su obtención económica y oportuna - con procesos de software.
- Diferenciar los conceptos de verificación y de validación de software, por medio de la aplicación de estándares y técnicas pertinentes a la evaluación de características de calidad de los productos, a fin de que puedan complementarse con las actividades de desarrollo o mantenimiento de software.
- Aplicar métodos de aseguramiento y control de la calidad del software, integrados a planes de proyectos, mediante la escogencia de procedimientos y técnicas apropiados, para una adecuada gestión y cobertura en la evaluación de productos de software.
- Examinar productos de software, mediante la aplicación de técnicas de evaluación apoyadas por herramientas, para la generación de mediciones informativas sobre su calidad.

4 Contenidos

- **Contexto de la Ingeniería del Software**
 - Origen del término 'Ingeniería del Software'
 - Evolución de la Ingeniería del Software
 - Cuerpo(s) de conocimientos de la Ingeniería del Software
 - Competencias profesionales en la Ingeniería del Software
 - Estándares de Ingeniería del Software
 - Educación de ingenieros de software
 - Certificaciones profesionales en Ingeniería del Software
 - Ética y conducta profesional en la Ingeniería del Software
- **Calidad del software**
 - Calidad del software
 - Fracazos de software
 - Requerimientos y calidad
 - Atributos de calidad
- **Procesos de software**
 - Concepto de proceso de software
 - Principales actividades en el desarrollo de software
 - Modelos de ciclos de vida: secuenciales, ciclo-V, iterativos, ágiles
 - Flujos de trabajo
 - Artefactos
 - Trabajadores y roles
 - Ejemplos documentados de procesos de software
 - Estándares de procesos de software
- **Control de la calidad y de sus costos**
 - Calidad y factores económicos
 - Defectos del software y terminología asociada
 - Costos de detección y de reparación de defectos
 - Distribución de los defectos en las actividades de desarrollo
 - Aseguramiento de la calidad; control de la calidad; verificación y validación de software
 - Prevención de defectos; detección temprana de defectos; contención de defectos
 - Responsabilidades en torno a la calidad
 - Beneficios de aplicar el aseguramiento de la calidad

- **Fundamentos de las pruebas de software**
 - ¿Qué es probar software?
 - ¿Por qué es necesario probar software?
 - Siete principios de las pruebas de software
 - Proceso de pruebas básico
 - La Psicología de las pruebas
 - Ética en las pruebas de software
- **Pruebas durante el ciclo de vida del software**
 - Niveles de prueba
 - Tipos de pruebas
 - Pruebas de mantenimiento
- **Verificación de software: Técnicas estáticas**
 - Conceptos y principios generales de las revisiones estáticas
 - Variedades de revisión estática
 - Procesos de revisión estática
 - Listas de cotejo
 - Estándares para revisiones de software y clasificación de anomalías de software
 - Aplicabilidad de las técnicas según los productos (artefactos) del proceso
 - Registro y reporte de datos de verificación
 - Relación con los procesos de desarrollo y de mantenimiento
- **Validación de software: Técnicas de pruebas**
 - Categorías de las técnicas de diseño de pruebas
 - Técnicas de pruebas funcionales, basadas en especificaciones ("caja negra")
 - Técnicas de pruebas funcionales, basadas en estructura ("caja blanca")
 - Técnicas basadas en la experiencia
 - Pruebas exploratorias
 - Especificación de pruebas
 - Aplicabilidad y selección de las técnicas de prueba
 - Otros tipos de pruebas: regresión, técnicas, integración, aceptación
 - Estándares para pruebas de software
 - Planes de pruebas
 - Ejecución de pruebas
 - Cobertura de las pruebas
 - Registro y reporte de datos de validación
 - Relación con los procesos de desarrollo y de mantenimiento
- **Herramientas de apoyo**
 - Consideraciones generales sobre las herramientas
 - Herramientas de análisis estático
 - Herramientas de ejecución de pruebas

- Herramientas de gestión de pruebas y procesos de verificación y validación
- Automatización de pruebas
- Uso efectivo de herramientas de pruebas: beneficios y riesgos potenciales
- **Gestión de las pruebas**
 - Organización de pruebas
 - Planificación y estimación de las pruebas
 - Seguimiento y control de las pruebas
 - Gestión de configuraciones de software
 - Riesgo y pruebas
 - Gestión de defectos e incidentes
- **Integración de la verificación y la validación a los procesos de software**
 - Sinergia de revisiones y pruebas
 - Ciclo V y principios de integración de V&V con procesos de software: secuenciales, iterativos, ágiles
 - Técnicas de V&V recomendadas por producto
 - Papeles y participación de los actores en los procesos de V&V en proyectos
 - Estándar para verificación y validación de software
 - Análisis de modos de falla
 - Valoración de riesgos para la verificación y la validación
 - Procesos ágiles para integración y liberación continuas (*DevOps*)
 - Estimación del esfuerzo dedicado a la verificación y la validación
 - Planificación de V&V (revisiones, pruebas y otros)
- **Aseguramiento de la calidad del software**
 - Actividades de gestión de la calidad en los proyectos informáticos
 - Componentes de apoyo al aseguramiento de la calidad del software
 - Estándar para procesos de aseguramiento de la calidad del software
 - Aseguramiento de la calidad en el mantenimiento de software
 - Aseguramiento de la calidad de las contribuciones externas
 - Integración del plan de calidad con el plan general del proyecto
 - Las funciones de aseguramiento y auditoría de la calidad
 - Análisis de causas raíz en problemas de calidad
 - Medición de la calidad
 - Agilidad y aseguramiento de la calidad
 - Alineamiento del aseguramiento de la calidad del software con el sistema organizacional de calidad

II parte: Aspectos operativos

5 Metodología de enseñanza y aprendizaje

Las lecciones serán impartidas de manera sincrónica, apoyadas en tecnologías por medio de la plataforma Zoom.

Se imparten clases magistrales, complementadas con ejemplos, casos de estudio, ejercicios y talleres de trabajo grupal. Los estudiantes investigarán sobre técnicas, estándares, procesos y herramientas de verificación y validación, elaborarán resúmenes explicativos sobre lo investigado – que serán discutidos dentro de los espacios habilitados en la clase sincrónica.

Se realizarán ejercicios y talleres de verificación y validación sobre modelos y componentes de software. Se realizarán trabajos extra-clase para aplicar revisiones, diseñar y especificar pruebas, ejecutar y automatizar pruebas, identificar riesgos y modos de falla, y desarrollar planes de verificación y validación.

Los temas abordados en clase deben ser complementados por el estudiante a través de lecturas recomendadas por el profesor. Esto dará al estudiante un mayor criterio acerca de los contenidos presentados en el curso, para un mejor desempeño en el ámbito profesional. Estas lecturas serán discutidas en clase y comprobadas durante la sesión inmediata a su asignación.

El curso cubre el sílabo (*syllabus*) de la certificación profesional internacional *Certified Tester – Foundation Level* del International Software Testing Qualifications Board (ISTQB)

Para la gestión de todos los aspectos de administración del curso: materiales, mecanismos de consulta por medio de chat o llamadas individuales y grupales, revisiones de proyectos y tareas en equipos se utilizará Microsoft Teams.

Se lleva el registro de participación en la lección sincrónica con el objetivo de determinar en cada lección los estudiantes faltantes y conectados. Esto por cuanto las plataformas por las que se desarrollan la sesión rastrean los espacios de conexión/desconexión de los participantes.

6 Evaluación

- Actividades: tareas / trabajos en clase
- Trabajo de investigación, exposición y participación en foros
- Un proyecto integrador

Los equipos de trabajo son de 5 personas máximo y no se permiten más de 6 equipos. Para todas las evaluaciones se realizará una rúbrica de evaluación de pares que tiene peso en la nota final individual, la cual puede hacer la diferencia en el resultado final de una evaluación grupal.

Cada equipo de trabajo recibirá un tema de investigación en la segunda semana del curso sobre el cual deberá entregar un documento formal de investigación y preparar una exposición detallada y confeccionar preguntas generadoras para un foro participativo que se contesta el día de la exposición antes de que acabe el día en que se expone. Los integrantes de la exposición no participan en su propio foro, pero deben participar en el foro del resto de equipos de manera obligatoria.

Rubro	Porcentaje
Un proyecto integrador	35%
Actividades semanales 12 de 3.75% c/u entre tareas y trabajos en clase	45%
Investigación 8% Exposición 7% Foros 5%	20%
Total	100%

7 Bibliografía

*Libros principales resaltados en **negrita**.*

- Amman, P. & Offutt, J. Introduction to Software Testing, 2nd ed. Cambridge University Press, 2016.
- Copeland, Lee. A practitioner's guide to software test design. Artech House, 2004.
- Craig, Rick; Jaskiel, Stefan. Systematic software testing. Artech House, 2002.
- Ebenau, R. G.; Strauss, S. H. Software inspection process. McGraw-Hill; 1994.
- Evans, I. Achieving Software Quality through Teamwork. Artech House, 2004.
- Forgács, I. & Kovács, A. Practical test design. British Computer Society, 2019.
- **Galin, Daniel. Software quality assurance. Pearson / Addison-Wesley, 2004.**
- Garbanzo, P. Procesos de desarrollo de software. Club de Investigación Tecnológica, 2005.
- Goericke, S. (ed.). The Future of Software Quality Assurance. Springer, 2020.
- Hambling B. Software Testing Foundation, 4th. Ed. British Computer Society, 2019.
- IEEE. IEEE Std. 730. IEEE Standard for Software Quality Assurance Processes. The Institute of Electrical and Electronics Engineers.
- IEEE. IEEE Std. 1028. IEEE Standard for Software Reviews. The Institute of Electrical and Electronics Engineers.
- IEEE. IEEE Std. 1044. IEEE Standard Classification for Software Anomalies. The Institute of Electrical and Electronics Engineers.
- ISO/IEC. ISO/IEC 9126, Software Engineering – Software Product Quality. Ginebra, Suiza: ISO/IEC.
- ISO/IEC. ISO/IEC 25000, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE.
- ISO/IEC. IEEE Std. 29119. IEEE Standard Software and systems engineering - Software testing. The Institute of Electrical and Electronics Engineers.
- ISTQB. Certified Tester - Foundation Level Syllabus. International Software Testing Qualifications Board, 2018.
- ISTQB. Standard Glossary of Terms Used in Software Testing Version 3.1. International Software Testing Qualifications Board, 2016.
- Jonassen Hass, A. Guide to Advanced Software Testing. Norwood, Estados Unidos: Artech House, 2008.
- **Jorgensen, P. Software Testing: A Craftsman's Approach, 4th. ed. Auerbach, 2013.**
- Kasse, Tim; McQuaid, Patricia A. Software Configuration Management for Project Leaders. Software Quality Professional 2(4) pp.8-19. American Society for Quality, 2000.
- Koomen, T. et al. TMap Next, for result-driven testing. CN's-Hertogenbosch, Países Bajos: UTN Publishers, 2006.
- Patton, Ron. Software testing, 2nd. Ed. SAMS, 2005.
- Sale, D. Testing Python. Wiley, 2014.
- **Spillner, Andreas et al. Software Testing Foundations - A Study Guide for the Certified Tester Exam, 4th ed. Rocky Nook, 2014.**
- Westfall, L. The Certified Software Quality Engineer Handbook, 2nd ed. American Society for Quality, Quality Press, 2017.
- egers, K. E. Peer reviews in software; a practical guide. Pearson / Addison-Wesley, 2002.

8 Profesor

Ericka Solano Fernández tiene estudios de Ingeniería en Computación (ITCR) y Maestría en Administración Educativa (Universidad Latina de Costa Rica). Actualmente cursa la Maestría en Gerencia de Tecnologías de Información del Tecnológico de Costa Rica.

Ha laborado para la Escuela de Computación como docente desde el año 2001 y actualmente se desempeña como docente de la Escuela de Computación en el Centro Académico de San José.

Ha impartido cursos como Fundamentos de Organización de Computadoras, Introducción a la Programación, Taller de Programación, Estructuras de Datos, Programación Orientada a Objetos, Lenguajes de Programación, Principios de Sistemas Operativos, Requerimientos de Software, Diseño de Software, Aseguramiento de la Calidad de Software, Análisis y Diseño de Algoritmos, Introducción al desarrollo de aplicaciones WEB, Elementos de Computación para la Escuela de Computación.

Ha impartido cursos de programación y metodologías de desarrollo de software en otras instituciones educativas desde el año 2006 y ha sido instructora en cursos de capacitación de actualización profesional en el sector público y privado.

Correo electrónico de contacto oficial: **ersolano@itcr.ac.cr**

Medio oficial electrónico: TEC-Digital (<http://tecdigital.tec.ac.cr>)

Grupo de comunicación en mensajería instantánea del curso (ver TEC Digital)

Horario de consulta digital:

Se establece el horario de consulta general para los cursos: **lunes de 9:30am a 11:00 am con reserva previa.**

Para otro horario específico deberá solicitarse por vía del correo electrónico, pues estaría sujeto a disponibilidad.